BESCHWERDEKAMMERN
DES EUROPÄISCHEN
PATENTAMTS

BOARDS OF APPEAL OF
THE EUROPEAN PATENT
OFFICE

CHAMBRES DE RECOURS
DE L'OFFICE EUROPÉEN
DES BREVETS

Internal distribution code:

(A) [-] Publication in OJ
(B) [-] To Chairmen and Members
(C) [-] To Chairmen
(D) [X] No distribution

Datasheet for the decision of 15 January 2020

Case Number:	T 0731/17 - 3.5.07
Application Number:	04779551.3
Publication Number:	1593059
IPC:	G06F17/30
Language of the proceedings:	EN

Title of invention:
Object persistence in a database store

Applicant:
Microsoft Technology Licensing, LLC

Headword:
Object persistence/MICROSOFT TECHNOLOGY LICENSING

Relevant legal provisions:
EPC Art. 56, 83, 84, 123(2)
RPBA 2020 Art. 11, 12(2)

Keyword:
Amendments - added subject-matter (no)
Claims - clarity (yes)
Sufficiency of disclosure - (yes)
Remittal to the department of first instance - (yes)

This datasheet is not part of the Decision.
It can be changed at any time and without notice.

Decisions cited:
G 0001/04, T 0641/00, T 0154/04, T 1954/08

This datasheet is not part of the Decision.
It can be changed at any time and without notice.

Europäisches Patentamt
European Patent Office
Office européen des brevets

Beschwerdekammern

Boards of Appeal

Chambres de recours

Boards of Appeal of the
European Patent Office
Richard-Reitzner-Allee 8
85540 Haar
GERMANY
Tel. +49 (0)89 2399-0
Fax +49 (0)89 2399-4465

Case Number: T 0731/17 - 3.5.07

DECISION
of Technical Board of Appeal 3.5.07
of 15 January 2020

Appellant:
(Applicant)

Microsoft Technology Licensing, LLC
One Microsoft Way
Redmond, WA 98052 (US)

Representative:

Grünecker Patent- und Rechtsanwälte
PartG mbB
Leopoldstraße 4
80802 München (DE)

Decision under appeal:

Decision of the Examining Division of the European Patent Office posted on 17 October 2016 refusing European patent application No. 04779551.3 pursuant to Article 97(2) EPC

Composition of the Board:

Chairman R. Moufang
Members: R. de Man
P. San-Bento Furtado

Summary of Facts and Submissions

I. The applicant (appellant) appealed against the decision of the Examining Division refusing European patent application No. 04779551.3, which was published as WO 2005/045707 and claims a priority date of 23 October 2003.

II. The decision was issued on EPO Form 2061 and referred for its reasons to the communication dated 2 May 2016. This communication cited the following documents:

D1: US 6 505 211 B1, published on 7 January 2003;

D2: E. Pardede et al.: "New SQL Standard for Object-Relational Database Applications", Proceedings of the 3rd IEEE Conference on Standardization and Innovation in Information Technology, 22-24 October 2003, Delft, The Netherlands, pp. 191-203;

D3: M. Carey et al.: "O-O, What Have They Done to DB2?", Proceedings of the 25th International Conference on Very Large Data Bases (VLDB), 7-10 September 1999, pp. 542-553;

D4: US 2002/0174128 A1, published on 21 November 2002.

The Examining Division decided that the sole substantive request did not comply with Articles 84 and 123(2) EPC and that the subject-matter of all claims 1 to 13 lacked an inventive step over a "notoriously known distributed computing environment comprising general purpose computers and a network". The decision also included a section arguing that the subject-matter of the independent claims was either known from or rendered obvious by both documents D1 and D2.

III. In its statement of grounds of appeal, the appellant maintained the sole substantive request considered in the contested decision.

IV. In a communication accompanying the summons to oral proceedings, the Board expressed the preliminary opinion that the sole substantive request did not comply with Articles 83, 84 and 123(2) EPC. It agreed with the appellant that the technical content of claim 1 went beyond a "notoriously known distributed computing environment comprising general purpose computers and a network" and that the Examining Division's inventive-step reasoning was therefore unconvincing.

V. In a letter dated 4 November 2019, the appellant replaced its sole request with a new sole main request.

VI. During oral proceedings held on 4 December 2019, the appellant replaced the main request with a new sole request. At the end of the oral proceedings, the chairman announced that the decision would be given in writing.

VII. The appellant requested that the decision under appeal be set aside and that a patent be granted on the basis of the following application documents:

- claims 1 to 5 of the sole request filed during the oral proceedings before the Board;
- description:
  - pages 2, 3 and 5 to 19 as published;
  - pages 4, 4a and 20 as filed with letter dated 30 April 2009;
  - page 1 as filed with letter dated 14 July 2011;
- drawings: sheets 1/8 to 8/8 as published.

VIII. Claim 1 of the sole request reads as follows:

"A method of executing a query on an object in a system in which the object is an instance of a user defined type that is persisted in a database store, wherein a definition of the user defined type comprises one or more fields and behaviors,

wherein at least one of the behaviors returns a value of one of the fields of the user defined type,

wherein each field of the user defined type is annotated with a first attribute that controls one or more storage facets of the field, wherein the first attribute in combination with an actual type of the field is used to control a storage layout of the value of the field, wherein the storage facets of the field comprise at least one of the maximum size of the field, whether or not the field is fixed length, the precision of the field, the scale of the field, and whether values of the field can be null,

wherein each of the at least one of the behaviors returning a value of the fields of the user defined type is annotated with a second attribute that denotes an equivalent structural access path, wherein the second attribute specifies a name of the field that is the subject of the behavior, wherein the name is used as the equivalent structural access path for the behavior,

wherein the object type is defined as a class in managed code, managed code being code running within a Common Language Runtime, CLR, environment, and

wherein the database store maintains information reflecting the storage layout as provided by the annotations to the type definition, the method comprising:

receiving (406), by a database server, a query that includes a predicate or an expression that references a behavior of the at least one of the behaviors returning a value of one of the fields of an object that is an instance of the user defined type;

accessing, by the database server, the information maintained by the database store to determine the storage layout of instances of the user defined type;

translating (408), by the database server, the query into the equivalent structural access path for a value of the field of the user defined type that is the subject of the referenced behavior;

structurally accessing (410), by the database server, the value in a table of the database store by parsing the object that is persisted in the database store without populating all parts of the object in the CLR environment and without invoking the referenced behavior of the object in the managed code; and

returning, by the database server, the value in response to the query."

Reasons for the Decision

1. The appeal complies with the provisions referred to in Rule 101 EPC and is therefore admissible.

2. *The invention*

2.1 The application relates to persisting objects in a data store. The background section explains that Microsoft SQL SERVER, which integrates the Microsoft Windows .NET Framework Common Language Runtime (CLR), allows creating a "user defined type" (UDT) class, instances of which can then be persisted in the database store (paragraphs [0003] and [0014] of the published application).

2.2 UDTs extend the scalar type system of the database and can be used in the same contexts as a system type, such as in column definitions, variables, parameters, function results, cursors, triggers, and replication (paragraph [0007]). The class that defines a UDT can include methods that implement specific behaviours on objects of that type (paragraph [0014]).

2.3 An object of a UDT class is persisted in the database store by a process known as "object serialisation", which transfers the values of the variables of the class to the database store's physical storage (paragraph [0015]).

When a database query which references a behaviour of a persisted UDT object is executed, the object has to be deserialised, memory for the full object has to be allocated in the CLR to receive the object's stored values, and the method implementing the behaviour has to be invoked on the full object (paragraph [0016]).

2.4 The invention aims to reduce the processing overhead associated with allocating memory for storing the full object at runtime, deserialising and populating all parts of the object, essentially by providing metadata that allows "direct structural access" to the field values in the serialised representation of the persisted object (paragraphs [0016] and [0017]).

3. *Added subject-matter - Article 123(2) EPC*

3.1 Present independent claim 1 corresponds to the combination of original claims 6 to 10, with a number of amendments taken from the description as discussed below.

3.2 At least one of the behaviours of the user defined type "returns a value of one of the fields of the user defined type" (as disclosed in paragraph [0036]).

3.3 The claim's "first attribute" corresponds to the "SqlUdtField" attribute shown in Figure 5, which "in combination with an actual type of the field is used to control a storage layout of the value of the field" (see paragraph [0039]).

3.4 The feature "managed code being code within a Common Language Runtime, CLR, environment" is based on paragraph [0004].

3.5 The received query "includes a predicate or an expression that references a behavior ... of an object that is an instance of the user defined type" (see paragraph [0041]).

3.6 As to the omission of the feature of original claim 6 specifying that "translating the query into an equivalent structural access path" is "based on the information about the storage layout of instances of the type", the Board notes that the "equivalent structural access path for a value of the field ... that is the subject of the referenced behavior" is denoted by the behaviour's "second attribute", which specifies the name of that field. The skilled person reading the application would therefore have understood that the translating step does not involve the "first attribute", which specifies information about the storage layout of the value of a field. This is consistent with paragraphs [0043] and [0045] of the description, which both disclose that the query is translated on the basis of the metadata associated with the UDT without specifically mentioning the first attribute.

3.7 The replacement of "without hydrating the object" with "by parsing the object that is persisted in the database store without populating all parts of the object in the CLR environment and without invoking the referenced behaviour of the object in the managed code" is based on paragraph [0044], which discloses that the value of the requested field is accessed structurally "without the need for object hydration and without invoking any behaviors in managed code" and is implemented by parsing the serialised form.

3.8 The value of the requested field is structurally accessed "by parsing the object that is persisted in the database store", without object hydration and "without invoking the referenced behavior of the object in the managed code" (paragraph [0044]). Paragraph

[0016] clarifies that object hydration in the CLR to allow invoking the behaviour on the full object involves "populating all parts of the object in the CLR environment".

3.9 The amendments made to claim 1 of the sole request therefore do not contravene Article 123(2) EPC.

4. *Clarity - Article 84 EPC*

4.1 In its decision, the Examining Division argued that claim 1 was not clear because the terms "storage facets" and "structural access path" had no generally agreed upon meaning in the art and were not defined in the claim.

4.2 In claim 1, the term "storage facet" is nothing more than a convenient label for storage-related properties of a field such as "the maximum size of the field, whether or not the field is fixed length, the precision of the field, the scale of the field, and whether values of the field can be null". Claim 1 states that a field's "first annotation" indicates one or more of such storage-related properties (i.e. "storage facets"). Hence, the term "storage facet" does not render claim 1 unclear.

4.3 As for the term "structural access path", claim 1 now clarifies that the "equivalent structural access path" associated with a behaviour that returns a value of a field of the user defined type is the name of that field.

4.4 The clarity objections raised by the Board in its communication no longer apply to claim 1 as now amended.

4.5 Hence, claim 1 of the sole request complies with Article 84 EPC.

5. *Sufficiency of disclosure - Article 83 EPC*

5.1 In its communication, the Board expressed doubts that the invention as claimed in then claim 1, in particular the step of "structurally accessing ... the value without de-serializing the object", was sufficiently disclosed within the meaning of Article 83 EPC.

These doubts were based on an interpretation of "structurally accessing ... the value without de-serializing the object" according to which the value of a field of an object is extracted from the serialised representation, not by parsing the serialised representation starting from its beginning, but by somehow directly accessing the portion of the serialised representation that contains the field's value.

5.2 In present claim 1, the feature "without de-serializing the object" has been replaced with "by parsing the object that is persisted in the database store without populating all parts of the object in the CLR environment and without invoking the referenced behavior of the object in the managed code".

Claim 1 therefore can no longer be interpreted as meaning that the field's value is extracted from the object's serialised representation by directly accessing the relevant portion and without parsing the representation from the beginning.

5.3 Instead, claim 1 now clarifies that the object persisted in the database corresponds to an object created in a CLR environment.

Paragraph [0016] of the background section of the application explains, with reference to Figure 2, that executing a database query referencing a behaviour of the persisted object would normally require the object to be recreated in the CLR environment to allow the referenced behaviour to be invoked on the object. Recreating the object involves de-serialising the object's serialised representation and reconstructing the full object in a memory area allocated within the CLR environment.

To eliminate this processing overhead in the specific case of a behaviour that returns the value of one of the fields of the object, claim 1 proposes providing the database server with information in the form of "first annotations" and "second annotations" that together allow the server, without having to involve the CLR environment, to parse the object's serialised representation and extract the field's value.

The Board has no reason to doubt that the skilled person could have put into practice the subject-matter of present method claim 1.

5.4 Thus, the invention as defined by claim 1 is sufficiently disclosed in the application (Article 83 EPC).

6. *The Examining Division's inventive-step reasoning*

6.1 In its decision, the Examining Division essentially argued that the only technical features of then claim 1 were "system", "database store", "server" and "persisted values" and that all other claim features, when taken in isolation, were non-technical because they were directed to a "non-technical rationale for providing 'more efficient storage and retrieval of objects persisted in a database store'".

It then argued, referring to point 6.2 of the reasons for decision T 1954/08 of 6 March 2013, that the non-technical features did not interact with the technical features to make a technical contribution because "effects stemming from the algorithmi[c] definition of a method do not define a technical character of the corresponding features".

Consequently, the subject-matter of claim 1 lacked inventive step over a "notoriously known distributed computing environment comprising general purpose computers and a network".

6.2 According to decision T 1954/08 of 6 March 2013, reasons 6.2, "the sole processing speed" of a computer-implemented algorithm and "the sole amount of memory" it requires are not suitable criteria for determining whether a method step contributes to the solution of a technical problem.

However, these statements in decision T 1954/08 cannot be taken to mean that any effect resulting from the implementation of a non-technical feature or combination of features is non-technical. If non-technical features could never contribute to a technical effect just because they are non-technical, there would be no need to analyse whether non-technical features interact with the technical subject-matter of the claim to solve a technical problem or bring about a technical effect, which would be contrary to opinion G 1/04 (OJ EPO 2006, 334), reasons 5.3, and decision T 154/04 (OJ EPO 2008, 46), reasons 5, under (F), and 13 to 15.

The Examining Division's analysis of the technical content of claim 1 is therefore flawed.

6.3 The Board further notes that the subject-matter of a claim lacks inventive step within the meaning of Article 56 EPC only if it can be shown that the skilled person, having regard to the state of the art, would have arrived at something that falls within the terms of the claim. The Examining Division should therefore have analysed whether the skilled person, starting from what it considered to be a suitable starting point in the prior art and faced with the objective technical problem, would indeed have arrived at a method comprising both the technical and the non-technical features of claim 1. In such an analysis, it is proper to include in the formulation of the technical problem non-technical features, not already part of the starting point or its technical context, only if those features make no technical contribution (see decision T 641/00, OJ EPO 2003, 352).

6.4 In the present case, any attempt to properly formulate a problem that potentially would have led the skilled person from a network of general-purpose computers to the subject-matter claimed should have confronted the Examining Division with the fact that the claim, not analysed as a collection of disconnected terms but as a whole, contains various technical concepts.

For example, the claimed method involves the concept of accessing information contained in a database store via a database server. Such technical functionality is not disclosed by a network of general-purpose computers. The Board is aware that database management systems were well known at the priority date of the application (see document D1, column 1, lines 27 to 30), but that does not mean that an inventive-step reasoning can silently ignore the concept.

6.5 The separate section of the decision discussing documents D1 and D2 contains no detailed analysis. If the Examining Division was of the view that document D1, in column 3, lines 56 to 62, discloses direct access to object attributes "without de-serialization" because the cited passage does not positively state that serialisation takes place, the Board observes that no disclosure of serialisation is not a disclosure of no serialisation.

6.6 In sum, the inventive-step reasoning contained in the contested decision is not convincing.

7. *Remittal*

7.1 In view of the application's filing date, the appellant requested the Board to exercise its discretion under Article 111(1) EPC to decide that the subject-matter now claimed is patentable.

7.2 The applicable version of the Rules of Procedure of the Boards of Appeal is the revised version that entered into force on 1 January 2020 (Articles 24 and 25(1) RPBA 2020). Under Article 11 RPBA 2020, a case is not to be remitted to the department whose decision was appealed unless special reasons present themselves for doing so. The Board notes that this provision has to be read in conjunction with Article 12(2) RPBA 2020, which provides that it is the primary object of the appeal proceedings to review the decision under appeal in a judicial manner.

7.3 The subject-matter of present claim 1, which represents a solution to a technical problem as discussed in point 5.3 above, is not rendered obvious by a network of general-purpose computers alone. However, inventive step over documents D1 to D4 has not yet been assessed in detail. Moreover, it may need to be investigated whether document D2 belongs to the state of the art under Article 54(2) EPC at all.

Not remitting the case to the Examining Division would require the Board to perform these tasks in both first- and last-instance proceedings and to effectively replace the Examining Division rather than review the contested decision in a judicial manner. It follows that special reasons within the meaning of Articles 11 and 12(2) RPBA 2020 present themselves.

7.4 Hence, the Board remits the case to the Examining Division for further prosecution on the basis of the sole request and recommends that it be dealt with expeditiously.

Order

For these reasons it is decided that:

1. The decision under appeal is set aside.

2. The case is remitted to the department of first instance for further prosecution.

The Registrar: The Chairman:

Beschwerdekammern des Europäischen Patentamts
Boards of Appeal of the European Patent Office
Chambres de recours de l'Office européen des brevets

I. Aperribay R. Moufang

Decision electronically authenticated